MUXMARK: MULTI-CHANNEL BIT MULTIPLEXING FOR LLM WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-bit LLM watermarking embeds recoverable messages for text provenance. With a fixed token budget and per-token bit load, larger payloads reduce coverage per message bit. Increasing this load improves coverage, but joint encoding based on disjoint vocabulary partitions concentrates the bias on exponentially smaller target regions, potentially weakening recovery signals at a fixed bias strength. We propose MuxMark, a multi-channel bit multiplexing framework that assigns selected message positions to separately keyed binary channels. During generation, channel contributions are summed to form a token-level logit bias that increases with the number of matched target bits. During decoding, the channels are reconstructed from the secret key and observed prefix, and weighted token votes are aggregated to recover the message bit by bit. Experiments across payloads, generation lengths, model families, and text perturbations demonstrate improved message recovery over existing multi-bit watermarking methods while maintaining comparable text quality, with particularly strong gains when recovering long messages from short texts.

1 INTRODUCTION

Large language models (LLMs) are increasingly used to generate text in real-world applications, including education, journalism, online platforms, and AI-assisted writing. As a result, machine-generated text is becoming deeply intertwined with human-written text, making it difficult to determine where a piece of text comes from. This growing indistinguishability creates serious concerns about misinformation, plagiarism, authorship attribution, and the large-scale misuse of synthetic content. Consequently, there is an urgent need for mechanisms that can reliably identify and trace machine-generated text after it has been released.

LLM Watermarking has emerged as one of the most promising directions for addressing this problem. Instead of modifying the final text after generation, LLM watermarking subtly biases the token sampling process such that the generated text contains a hidden statistical pattern. Existing work (Kirchenbauer et al., 2023) has shown that such patterns can be detected with little impact on text quality, and recent studies (Yoo et al., 2024; Qu et al., 2025; Feng et al., 2025; Jiang et al., 2025b) have further extended watermarking from simple binary detection (zero-bit watermarking) to multi-bit message embedding, where the generated text carries a hidden message such as a model identifier, user ID, or generation source.

Compared with zero-bit detection, multi-bit watermarking requires sufficient statistical evidence to recover every position of an embedded message. This requirement becomes particularly challenging under a limited token budget. Representative methods assign each token to a message position or segment and recover the message by aggregating repeated observations (Yoo et al., 2024; Qu et al., 2025; Feng et al., 2025). As shown in Figure 1, when the number of bits assigned to each token is held fixed, increasing the message payload distributes these observations across more message positions, reducing the average coverage per bit. Maintaining adequate coverage therefore motivates assigning more message bits to each token. The central challenge is how to increase this token-level load while retaining sufficiently reliable evidence for message recovery.

Assigning more bits to each token increases coverage, but reliable recovery also depends on how these bits influence token selection. MPAC (Yoo et al., 2024), whose position-allocation framework has informed subsequent multi-bit watermarking methods, jointly encodes k bits by partitioning the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

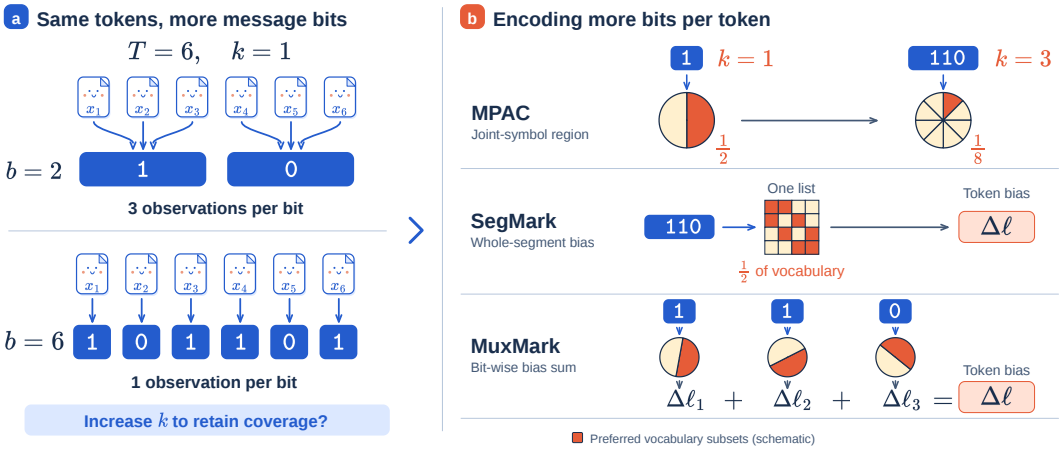


Figure 1: (a) Larger payloads reduce observations per bit at fixed T and k. (b) Increasing k narrows MPAC's target region, SegMark uses segment-level lists, while MuxMark sums channel-wise biases.

vocabulary into 2^k disjoint regions and biasing the target region. Under balanced random partitioning, this region receives a 2^{-k} share of the original probability mass in expectation. As k grows, the shrinking target can weaken the recovery signal at a fixed bias strength. SegMark (Qu et al., 2025) avoids this shrinkage by mapping each message-segment value to a green list covering half of the vocabulary. Both approaches organize encoding around the joint value of the selected bits. We investigate whether composing the token-level preference from individual bit contributions can improve message recovery under limited token budgets.

To this end, we propose MuxMark, a multi-channel bit multiplexing framework for reliable multi-bit message recovery. For each token, MuxMark assigns the selected message positions to separately keyed binary channels, each of which defines its own vocabulary bipartition. During generation, the channel contributions are summed to form the token-level logit bias, giving higher watermark scores to tokens that match more of the selected message bits. The resulting encoding preference reflects the contributions of individual bits, rather than a single reward for matching the complete target symbol. During decoding, the observed token provides a binary observation in each reconstructed channel, and its weighted vote is accumulated at the corresponding message position. Aggregating these votes across the text recovers the message bit by bit. This construction allows a shared token to contribute separately evaluated evidence to multiple message positions.

We examine how channel multiplicity affects bit-level observations and message recovery under different payloads and generation budgets. Experiments across various message lengths, generation lengths, model families, and text perturbations show that MuxMark consistently improves message recovery over existing multi-bit watermarking methods while maintaining comparable text quality. The improvement is particularly pronounced when long messages must be recovered from short generations. These results support multi-channel bit multiplexing as an effective way to improve message recovery under limited token budgets.

2 Related Works

Zero-bit Watermarking. The pioneering KGW method (Kirchenbauer et al., 2023) partitions the vocabulary into "green" and "red" token lists and biases green-token logits during generation to embed a detectable signal. Subsequent work improves robustness through globally fixed partitions (Zhao et al., 2024) or semantic, frequency-based, and context-aware features (Liu et al., 2024; Ren et al., 2024; Hou et al., 2024b;a). More recently, DualGuard (Li et al., 2026) combines complementary watermark streams to defend against both paraphrasing and spoofing. To mitigate quality degradation, entropy-aware schemes (Lu et al., 2024; Lee et al., 2024; Wang et al., 2025) adapt watermark strength to generation uncertainty, while unbiased or distortion-free methods (Hu et al., 2024; Chen et al., 2025; Mao et al., 2025) aim to preserve the original sampling distribution. Recent partitioning

schemes further exploit model probabilities: WaterMod (Park et al., 2026) partitions tokens by modular probability ranks, while SSG (Gu et al., 2026) constructs logit-balanced subsets to improve detectability in low-entropy settings. SynthID-Text (Dathathri et al., 2024) demonstrates scalable deployment with low generation overhead and efficient detection. Other recent extensions address reasoning models (Liu et al., 2026a), in-context watermarking (Liu et al., 2026b), semantic-level watermarking (Huo et al., 2026), and diffusion language models (Gloaguen et al., 2026).

Multi-bit Watermarking. Multi-bit methods encode explicit provenance information through message-dependent watermark construction (Fernandez et al., 2023; Jiang et al., 2025a; Cohen et al., 2025). Wang et al. (2024) enumerate candidate messages during decoding incur costs that grow exponentially with payload length. MPAC (Yoo et al., 2024) improves decoding efficiency through hash-based message allocation and position-wise voting. Building on this paradigm, SegMark (Qu et al., 2025) combines balanced allocation with error-correcting codes to improve recovery under perturbations. BiMark (Feng et al., 2025), StealthInk (Jiang et al., 2025b), and subsequent schemes (Jiang et al., 2026; Wang et al., 2026) pursue multi-bit embedding while preserving the original sampling distribution. More recently, XMark (Xu et al., 2026) combines leave-one-shard-out encoding, intersecting green lists, and constrained voting to improve recovery from short texts. QuantileMark (Zhu et al., 2026) partitions the cumulative probability interval into equal-mass bins to reduce message-dependent variation in embedding and decoding. HeRo (Chen et al., 2026) enables selective payload disclosure through hierarchical vocabulary routing, while BREW (Kim et al., 2026) separates message estimation from window-shifting verification to control false positives.

3 PRELIMINARIES

Notations. Consider an LLM Θ, $\mathcal{V}$ denotes the vocabulary with size $|\mathcal{V}|$. Given a prompt p_0 and contextual tokens $\mathbf{x}_{<t} = (x_1, x_2, \ldots, x_{t-1})$, the LLM generate an imminent token $x_t \in \mathcal{V}$ sequentially from the conditional distribution $P_\Theta(x_t \mid p_0, \mathbf{x}_{<t})$.

Bit-allocation Paradigm. This paradigm assigns message positions to generated tokens and encodes message bits via KGW (Kirchenbauer et al., 2023), having two steps: *(1) Message Encoding.* Let $\mathbf{m} \in \sum^b$ denote the message, where $\sum = \{0, 1\}$ is a binary alphabet and b is the message length. Let k denote the number of bits each token is allocated, these methods divide the whole message into $\tilde{b} = \lceil \frac{b}{k} \rceil$ segments. At each generation step t, it employs a PRF with a random seed to sample a segment position $\tilde{p} \in \{0, \ldots, \tilde{b} - 1\}$ and retrieve its message segment $\mathbf{m}[\tilde{p}]$. Meanwhile, the vocabulary is randomly shuffled and partitioned into 2^k disjoint subsets $\mathcal{V}_t = [C_0, ..., C_j, ..., C_{2^k-1}]$. According to the value of $\mathbf{m}[\tilde{p}]$, the corresponding subset is designated as the green list and is assigned a positive logits bias δ, which increases the likelihood that LLM samples tokens from that subset. *(2) Message Decoding.* Extract the message via a majority voting matrix $\mathbf{M} \in \mathbb{R}^{\tilde{b} \times 2^k}$ initialized to zero. For each token x_t, the decoder reconstructs the segment position $\tilde{p}$ and the vocabulary partitions $\mathcal{V}_t$ using the same PRF as in the encoding stage. If $x_t \in C_j$, the corresponding entry $\mathbf{M}[\tilde{p}][j]$ is incremented by one. After processing the entire text, each message segment is recovered via $\hat{\mathbf{m}}[\tilde{p}] = \arg\max_{j \in \{0,\ldots,2^k-1\}}(\mathbf{M}[\tilde{p}][j])$. Under the assumption that watermarked text contains more green tokens than non-green tokens, the aggregate majority count $\sum_{\tilde{p}=0}^{\tilde{b}-1} \max_j(\mathbf{M}[\tilde{p}][j])$ serves as the total green tokens for computing the z-score, which determines whether the text is watermarked.

4 MUXMARK

MuxMark embeds messages through multi-channel bit multiplexing, allowing each token to contribute evidence to multiple message positions. At each generation step, it assigns selected positions to separately keyed binary channels and sums their contributions into a token-level logit bias, favoring tokens that match more channels. During decoding, it reconstructs the assignments and channels from the secret key and observed prefix, then aggregates weighted token votes at the corresponding positions to recover the message bit by bit.

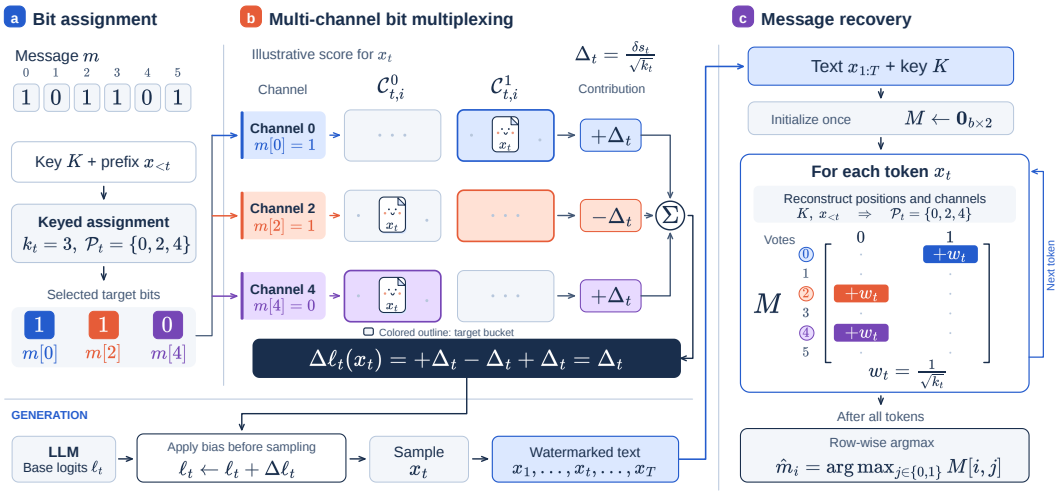


Figure 2: Overview of MuxMark.

4.1 Stochastic Bit Assignment

Let $\mathbf{m} \in \{0,1\}^b$ denote a b-bit message, where b is the message length. At each generation step t, MuxMark first determines the number of message positions assigned to the current token. Specifically, we define

$$k_t = 1 + \left(\mathrm{PRF}_K^{\mathrm{num}}(x_{<t}) \bmod k_{\max}\right), \tag{1}$$

where K is the secret watermark key, $x_{<t}$ is the token prefix, and $k_{\max}$ is the maximum number of message positions assigned to one token.

After obtaining k_t, we use another domain-separated pseudorandom function to sample k_t distinct positions from the message:

$$\mathcal{P}_t = \mathrm{Sample}_K\left(x_{<t}, b, k_t\right) = \left\{p_t^1, \ldots, p_t^{k_t}\right\}, \tag{2}$$

where $p_t^i \in \{0, \ldots, b-1\}$ denotes the message position assigned to the i-th channel of token x_t.

This stochastic assignment distributes token-level observations across different message positions and avoids repeatedly using a fixed allocation pattern. Importantly, both k_t and $\mathcal{P}_t$ depend only on the secret key and the observed prefix. The decoder can therefore reconstruct the same assignments from the watermarked text without accessing the generating model, the original prompt, or the next-token probability distribution.

4.2 Multi-Channel Bit Multiplexing

The preceding assignment allows each token to cover multiple message positions, but this coverage alone does not ensure reliable recovery. We address this through multi-channel bit multiplexing, which encodes the selected bits through separate binary channels without concentrating the encoding preference on a single target-symbol region. Channel contributions are summed in the token logits, allowing individual bit matches to contribute to the watermark score and giving higher scores to tokens that support more target bits.

Concretely, suppose token x_t is assigned to k_t message positions $\mathcal{P}_t = \left\{p_t^1, p_t^2, \ldots, p_t^{k_t}\right\}$. For the i-th selected position p_t^i and its corresponding channel index r_t^i, we use a keyed pseudorandom function to construct a channel-specific binary vocabulary partition

$$\mathcal{V}_{t,i} = \left\{\mathcal{C}_{t,i}^0, \mathcal{C}_{t,i}^1\right\}. \tag{3}$$

According to the target message bit $m[p_t^i] \in \{0,1\}$, the corresponding bucket $\mathcal{C}_{t,i}^{m[p_t^i]}$ is designated as the target set of channel i. Different channels use separately keyed binary partitions, so the membership of a token in one channel does not determine its membership in another.

During generation, we evaluate how many target channels are matched by the current candidate token x_t. Specifically, we define

$$h(x_t) = \sum_{i=1}^{k_t} \mathbb{I}\left[x_t \in \mathcal{C}_{t,i}^{m[p_t^i]}\right]. \tag{4}$$

Here, $h(x_t) \in \{0, \ldots, k_t\}$ measures the degree to which x_t agrees with the selected message bits. A token with $h(x_t) = k_t$ matches the complete target pattern, whereas a token with $0 < h(x_t) < k_t$ still supports a subset of the target bits. This partial agreement is discarded by conventional joint partitioning but is explicitly preserved by our multi-channel construction.

To combine the channel signals without allowing the total watermark strength to grow linearly with k_t, we scale the per-channel bias as

$$\Delta_t = \frac{\delta s_t}{\sqrt{k_t}}, \tag{5}$$

where δ is the base watermark strength and s_t is the entropy-based scaling factor,

$$s_t = s_{\min} + (s_{\max} - s_{\min})\sigma\left(\frac{H_t - c}{\tau}\right), \qquad H_t = -\sum_{x \in \mathcal{V}} P_t(x) \log P_t(x). \tag{6}$$

The resulting logit modification for candidate token x_t is

$$\Delta\ell_t(x_t) = -k_t\Delta_t + 2\Delta_t h(x_t), \tag{7}$$

which can be interpreted as a superposition of k_t binary channel signals. For each channel, matching the target bit contributes $+\Delta_t$, whereas failing to match it contributes $-\Delta_t$. Consequently, the logit modification increases progressively with the number of matched channels:

$$-k_t\Delta_t,\ -(k_t-2)\Delta_t,\ \ldots,\ (k_t-2)\Delta_t,\ k_t\Delta_t. \tag{8}$$

This graded structure is the key difference from joint multi-bit encoding. A joint 2^{k_t}-way partition rewards only the vocabulary region corresponding to the complete target pattern. In contrast, multi-channel bit multiplexing preserves the contribution of every correctly matched bit. Therefore, even when the sampled token does not satisfy all k_t channels, its matched channels can still provide valid bit-level observations during message recovery.

4.3 Channel-wise Message Recovery and Detection

Bit-level Message Recovery. For token x_t,we use the prefix $x_{<t}$ and the same PRF to reconstruct the channel assignments $\{(p_t^i, r_t^i)\}_{i=1}^{k_t}$, where p_t^i denotes the i-th message position encoded by x_t, and r_t^i denotes the corresponding binary channel index. For each channel r_t^i, if x_t falls into bucket $C_{t,i}^j$ with $j \in \{0,1\}$, update the bit-level voting matrix by

$$\mathbf{M}[p_t^i][j] \leftarrow \mathbf{M}[p_t^i][j] + w_t\mathbb{I}[x_t \in C_{t,i}^j], j \in \{0,1\}, \tag{9}$$

where w_t denotes the normalized voting strength of x_t according to the number of bits it carries. We set $w_t = \frac{1}{\sqrt{k_t}}$, so that tokens carrying more channels contribute multiple observations, while each observation is down-weighted to prevent large-k_t tokens from dominating the voting matrix. After processing all tokens, each bit position is decoded separately by $\hat{\mathbf{m}}[p] = \arg\max_{j\in\{0,1\}}(\mathbf{M}[p][j])$. The final recovered message is obtained by concatenating all decoded bits.

Message-conditioned Soft Detection. For each scored token x_t, we measure agreement between its channel observations and the recovered message $\hat{\mathbf{m}}$:

$$\mathrm{hit}_t = \sum_{i=1}^{k_t} \mathbb{I}\Big[x_t \in C_{t,i}^{\hat{\mathbf{m}}[p_t^i]}\Big], \qquad a_t = \frac{\mathrm{hit}_t}{k_t}. \tag{10}$$

The fractional agreement a_t allows a token to contribute partial evidence when only a subset of its channels agrees with the recovered message. For a text containing T scored tokens, we define the normalized detection score as

$$z = \frac{\sum_t a_t - \frac{T}{2}}{\sqrt{\frac{1}{4}\sum_t \frac{1}{k_t}}}. \tag{11}$$

The centering and scaling terms provide a reference normalization. Since channel observations may be dependent and the message is recovered from the same text, we use z as an empirical detection score without assuming a standard-normal null distribution.

For each evaluation setting, we sort the detection scores of the labeled watermarked and unwatermarked samples and consider the midpoints between adjacent distinct scores as candidate thresholds. We select the threshold that maximizes the F1 score on these samples among the candidates. A text is classified as watermarked if z larger than the threshold, and as unwatermarked otherwise.

5 Experiments

Overview. We evaluate MuxMark along three key dimensions: **payload**, **text quality**, and **robustness**. For payload, we test whether MuxMark can reliably detect watermark and recover messages. For text quality, we measure both perplexity and downstream task performance across representative generation tasks. For robustness, we evaluate detectability under realistic editing, copy-paste, and paraphrasing attacks.

5.1 Experimental Setup

Baselines. We compare MuxMark with the SOTA HeRo (Chen et al., 2026), latest XMark (Xu et al., 2026), unbiased BiMark (Feng et al., 2025), StealthInk (Jiang et al., 2025b), classic MPAC (Yoo et al., 2024), and multi-position SegMark (Qu et al., 2025). Message lengths are set to 16, 24, 32, 48 bits, we even test performance up to 512 bits. Detailed settings are in Appendix A.

Datasets. Following Kirchenbauer et al. (2023); Zhao et al. (2024), we randomly sample 200 prompts from the C4 (Raffel et al., 2020) dataset for text generation. We further evaluate downstream performance on text summarization and question answering tasks. More details are in Appendix B.

Metrics. We primarily use Bit Accuracy and Exact-match Accuracy to evaluate message decoding correctness. Detectability is measured using AUC score, while text quality is assessed with Perplexity (PPL), ROUGE-L, and the GPT5-Score. Metric definitions are provided in Appendix C.

Models. We use LLaMA3.1-8B-IT (Grattafiori et al., 2024) and Gemma3-12B-IT (Team, 2025). For fairness, we compute PPL using the larger Vicuna-13B-v1.5 model (Chiang et al., 2023).

Implementation Details. All experiments were implemented in Python 3.12.3 with PyTorch 2.8.0, CUDA 12.8 and two NVIDIA V100 32 GB GPU. We set the green-list ratio to $\gamma = 0.5$, the watermark strength is set to $\delta = 1.9$, and the maximum number of bits carried by each token is $k_{\max} = 6$. The number of binary channels is set equal to the message length. We use a context window size of 2 with the `lefthash` seeding scheme. For entropy factor, we set $s_{\min} = 0.4$, $s_{\max} = 1.3$, $c = 0.5$, and $\tau = 0.1$. The scaling factors are normalized within each batch and clipped to $[0.4, 1.3]$. During generation, we apply the watermark bias only to the top-10 candidate tokens.

5.2 Message Embedding payload

Multi-bit watermarking scenario. We evaluate MuxMark on two LLMs under different message lengths and generation budgets, as shown in Table 1. Overall, MuxMark achieves the highest Bit Acc in 15 out of 16 settings, improving the average Bit Acc from 0.899 of the strongest baseline

Table 1: Comparison of multi-bit watermarking performance.

Bits	Watermark	LLAMA3.1-8B-IT						GEMMA3-12B-IT					
		MAX 128 TOKENS			MAX 256 TOKENS			MAX 128 TOKENS			MAX 256 TOKENS		
		AUC	Bit Acc	Exact Acc	AUC	Bit Acc	Exact Acc	AUC	Bit Acc	Exact Acc	AUC	Bit Acc	Exact Acc
16	MPAC	0.998	0.944	0.405	0.999	0.982	0.755	0.921	0.833	0.030	0.972	0.899	0.265
	SegMark	0.987	0.893	0.595	0.999	0.987	0.940	0.482	0.503	0.000	0.508	0.512	0.000
	BiMark	0.999	0.971	0.635	**1.000**	0.996	0.945	0.985	0.867	0.125	0.998	0.937	0.450
	StealthInk	0.980	0.907	0.250	0.999	0.952	0.475	0.393	0.513	0.000	0.504	0.498	0.000
	XMark	0.996	0.940	0.460	**1.000**	0.987	0.860	0.839	0.809	0.095	0.918	0.892	0.275
	HeRo	**1.000**	0.981	0.805	**1.000**	0.996	0.955	0.983	0.852	0.230	0.998	0.931	0.475
	MuxMark	**1.000**	**0.988**	**0.840**	**1.000**	**0.999**	**0.990**	**0.998**	**0.880**	**0.605**	**0.999**	**0.991**	**0.885**
24	MPAC	0.993	0.894	0.075	0.999	0.962	0.405	0.898	0.786	0.000	0.968	0.856	0.040
	SegMark	0.915	0.747	0.230	0.997	0.973	0.885	0.498	0.504	0.000	0.483	0.497	0.000
	BiMark	**1.000**	0.932	0.220	**1.000**	0.983	0.695	0.973	0.802	0.005	0.996	0.891	0.155
	StealthInk	0.965	0.843	0.015	0.995	0.921	0.205	0.526	0.503	0.000	0.583	0.507	0.000
	XMark	0.981	0.864	0.065	0.995	0.959	0.485	0.826	0.758	0.005	0.872	0.824	0.040
	HeRo	**1.000**	0.962	0.535	**1.000**	0.994	**0.920**	0.983	0.810	0.045	**0.999**	0.888	0.165
	MuxMark	0.999	**0.974**	**0.545**	0.999	**0.995**	**0.920**	**0.999**	**0.930**	**0.245**	**0.999**	**0.973**	**0.560**
32	MPAC	0.993	0.858	0.020	**1.000**	0.942	0.155	0.823	0.754	0.000	0.968	0.828	0.015
	SegMark	0.868	0.689	0.065	0.994	0.921	0.640	0.500	0.491	0.000	0.525	0.492	0.000
	BiMark	0.999	0.882	0.025	**1.000**	0.965	0.405	0.979	0.751	0.000	0.991	0.845	0.025
	StealthInk	0.948	0.808	0.000	0.993	0.894	0.045	0.443	0.501	0.000	0.499	0.511	0.000
	XMark	0.981	0.829	0.005	0.999	0.921	0.170	0.810	0.709	0.000	0.859	0.795	0.005
	HeRo	**1.000**	0.945	**0.375**	**1.000**	0.968	0.680	0.989	0.785	0.015	**0.997**	0.859	0.025
	MuxMark	0.999	**0.949**	0.205	**1.000**	**0.987**	**0.700**	**0.998**	**0.903**	**0.050**	**0.997**	**0.962**	**0.355**
48	MPAC	0.977	0.808	0.000	0.999	0.890	0.000	0.865	0.710	0.000	0.938	0.766	0.000
	SegMark	0.760	0.606	0.000	0.957	0.764	0.125	0.519	0.501	0.000	0.506	0.491	0.000
	BiMark	**1.000**	0.777	0.000	**1.000**	0.926	0.040	0.965	0.644	0.000	0.989	0.780	0.000
	StealthInk	0.927	0.760	0.000	0.993	0.843	0.000	0.505	0.502	0.000	0.608	0.508	0.000
	XMark	0.943	0.756	0.000	0.995	0.876	0.020	0.757	0.674	0.000	0.821	0.748	0.000
	HeRo	**1.000**	0.902	**0.025**	**1.000**	**0.969**	0.140	0.970	0.736	0.000	0.995	0.810	0.005
	MuxMark	**1.000**	**0.911**	0.010	**1.000**	0.966	**0.240**	**0.976**	**0.851**	0.000	**0.998**	**0.918**	**0.070**

HeRo to 0.949, with an average absolute gain of 4.9 percentage points. The average gains over HeRo are 0.7 percentage points on LLaMA3.1-8B-IT and 9.2 percentage points on Gemma3-12B-IT, demonstrating improvements across both models. Under the most payload-constrained setting of 48-bit messages with a 128-token generation budget, MuxMark improves Bit Acc over the best baseline by 0.9 and 11.5 percentage points on LLaMA3.1-8B-IT and Gemma3-12B-IT, respectively. These results demonstrate effective message recovery under limited token budgets, with particularly pronounced gains on Gemma3-12B-IT.

Zero-bit watermarking scenario. As shown in Table 1, MuxMark achieves the best or tied-best AUC in 13 out of 16 settings, falling only 0.001 below the best baseline in each of the remaining three settings. On average, MuxMark obtains an AUC of 0.998, exceeding the strongest baseline HeRo at 0.995. AUC and Bit Acc capture different aspects of watermarking performance: AUC measures the ability to distinguish watermarked from unwatermarked text, whereas Bit Acc measures the correctness of the recovered message bits. These results show that MuxMark improves multi-bit message recovery while maintaining strong watermark detectability.

5.3 TEXT QUALITY PRESERVATION

Perplexity (PPL). Watermarking performance should be evaluated jointly in terms of text quality and message recovery. Figure 3 plots Bit Accuracy against watermarked PPL, varying δ for methods with tunable strength. HeRo is represented by a single point because its watermark strength is fixed. MuxMark generally achieves higher Bit Accuracy at comparable PPL. Notably, it attains approximately 0.91 Bit Accuracy at PPL 11.5, compared with HeRo's 0.90 at PPL 25.0. These results demonstrate a favorable trade-off between text quality and decoding accuracy, with δ allowing MuxMark to accommodate different operating requirements.

Downstream task performance. We further evaluate task-level utility on summarization and long-form QA, as shown in Table 2. For summarization, MuxMark reduces average ROUGE-L by 0.0013, compared with an average reduction of 0.0028 across baselines. For long-form QA, the paired

Figure 3: Bit Acc under PPL with various δ.

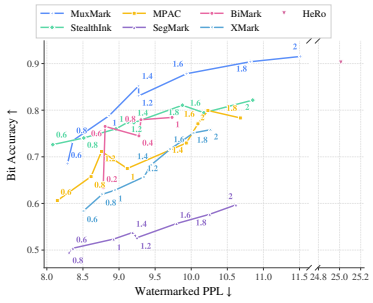


Table 2: Downstream task performance with 95% CIs.

Model	*Task 1: Text Summarization*			*Task 2: Long-form QA*		
+ Watermark	AUC	Bit Acc	ROUGE-L ↑ [95% CI]	AUC	Bit Acc	Δ GPT5-Score ↑ [95% CI]
LLAMA3.1-8B-IT	-	-	0.152	-	-	–
+ MPAC	0.933	0.869	0.1468 [0.1393, 0.1547]	0.987	0.857	-0.008 [−0.220, 0.205]
+ SegMark	0.910	0.757	0.1460 [0.1388, 0.1532]	0.900	0.741	+0.052 [−0.167, 0.272]
+ BiMark	0.964	0.824	0.1508 [0.1433, 0.1587]	0.994	0.881	-0.091 [−0.312, 0.130]
+ StealthInk	0.870	0.813	**0.1540** [0.1467, 0.1615]	0.865	0.775	**+0.079** [−0.140, 0.303]
+ XMark	0.871	0.821	0.1496 [0.1418, 0.1576]	0.947	0.836	+0.012 [−0.233, 0.259]
+ HeRo	**0.980**	0.915	0.1320 [0.1247, 0.1397]	**1.000**	0.874	-0.472 [−0.743, −0.203]
+ **MuxMark**	0.964	**0.931**	0.1517 [0.1435, 0.1600]	**1.000**	**0.971**	-0.446 [−0.708, −0.183]
GEMMA3-12B-IT	-	-	0.214	-	-	–
+ MPAC	0.526	0.583	0.2113 [0.2033, 0.2197]	0.885	0.742	-0.049 [−0.161, 0.065]
+ SegMark	0.520	0.488	0.2122 [0.2040, 0.2208]	0.501	0.506	**+0.014** [−0.098, 0.123]
+ BiMark	0.540	0.476	0.2162 [0.2071, 0.2255]	0.933	0.736	+0.004 [−0.109, 0.116]
+ StealthInk	0.502	0.544	**0.2168** [0.2086, 0.2252]	0.678	0.660	-0.056 [−0.180, 0.063]
+ XMark	0.513	0.564	0.2140 [0.2060, 0.2222]	0.789	0.703	0.000 [−0.138, 0.135]
+ HeRo	**0.648**	0.573	0.2132 [0.2049, 0.2217]	0.960	0.711	+0.011 [−0.103, 0.123]
+ **MuxMark**	0.625	**0.668**	0.2117 [0.2034, 0.2203]	**0.997**	**0.903**	-0.087 [−0.221, 0.047]

Table 3: Robustness evaluation under different attack settings. Results are reported for a 16-bit message with a maximum of 256 tokens on C4 using LLaMA3.1-8B-IT.

Watermark	**Word-D**		**Word-S-DICT**		**Word-S-BERT**		**Copy-Paste**		**Translation**		**Doc-P-Dipper-1**		**Doc-P-Dipper-2**	
	AUC	Bit Acc	AUC	Bit Acc	AUC	Bit Acc	AUC	Bit Acc	AUC	Bit Acc	AUC	Bit Acc	AUC	Bit Acc
MPAC	0.995	0.910	0.986	0.889	0.988	0.893	0.997	0.933	0.499	0.596	0.979	0.900	0.968	0.878
SegMark	0.949	0.823	0.936	0.787	0.940	0.806	0.975	0.879	0.506	0.547	0.943	0.789	0.928	0.773
BiMark	0.999	0.951	0.999	0.933	0.999	0.943	**1.000**	0.976	0.587	0.517	**0.999**	0.946	**0.996**	0.931
StealthInk	0.880	0.799	0.829	0.789	0.889	0.803	0.965	0.872	0.477	0.556	0.918	0.842	0.907	0.819
XMark	0.905	0.898	0.959	0.865	0.959	0.879	0.983	0.912	0.027	0.583	0.947	0.873	0.925	0.857
HeRo	**1.000**	0.954	0.999	0.952	**1.000**	0.918	**1.000**	0.984	**0.649**	0.588	0.972	0.880	0.979	0.887
MuxMark	**1.000**	**0.981**	**1.000**	**0.968**	0.999	**0.970**	**1.000**	**0.988**	0.614	**0.637**	0.990	**0.964**	0.988	**0.947**

evaluation reveals model-dependent quality effects: MuxMark exhibits a statistically significant score decrease on LLaMA3.1-8B-IT ($\Delta = -0.446$), whereas no significant change is detected on Gemma3-12B-IT ($\Delta = -0.087$), based on the reported 95% CIs. MuxMark achieves the highest Bit Accuracy across all four settings, including gains of 9.0 and 16.1 percentage points over the best baseline in QA on the two models, respectively. These results demonstrate stronger message recovery with small summarization losses, while revealing a QA quality trade-off on LLaMA3.1-8B-IT.

5.4 Robustness Results

We evaluate three common attacks: editing (word deletion and synonym substitution), copy-paste, translation, and Dipper paraphrasing. Full settings are in Appendix D. Table 3 shows that MuxMark achieves the highest Bit Acc under all attack settings, which improves Bit Acc over the best baseline in each setting by 2.3 percentage points on average. Translation remains the most challenging attack, causing substantial degradation in both AUC and Bit Acc across methods. MuxMark still achieves the highest Bit Acc of 0.637, exceeding the best baseline by 4.1 percentage points. Under Dipper paraphrasing, MuxMark also achieves the highest Bit Acc, improving over BiMark by 1.8 and 1.6 percentage points under the two settings, respectively. Its slightly lower AUC than BiMark further illustrates that stronger detectability does not necessarily imply more accurate message recovery.

6 Further Analysis

Ablation study. Table 4 reports ablations on C4 using LLaMA3.1-8B-IT with 48-bits and max-256-tokens. Replacing multi-channel encoding with a conventional 2^{k_t}-way partition, with allocation unchanged, reduces Bit Acc from 0.966 to 0.681 and AUC from 1.000 to 0.775. Fixed allocation ($k = 6$) and removing entropy scaling reduce Bit Acc to 0.952 and 0.963, respectively, while retaining an AUC of 1.000. Thus, multi-channel encoding provides the main gain in this setting.

Table 4: Ablation study of MuxMark.

Method	AUC	Bit Acc.
MuxMark (Full)	1.000	0.966
w/ Conventional 2^{k_t}-way Partition	0.775	0.681
w/ Fixed Allocation ($k = 6$)	1.000	0.952
w/o Entropy Scaling	1.000	0.963

Long messages with short outputs. Figure 4 (a) examines the challenging regime in which only $T = 64$ tokens are available for embedding messages of $b \in \{64, 128, 256\}$ bits. As the message

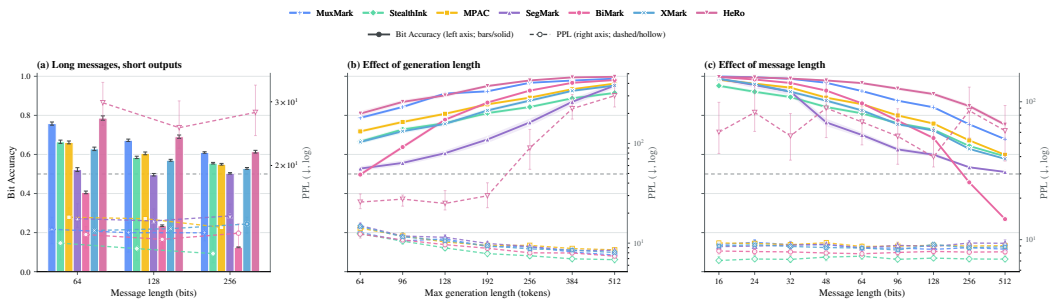


Figure 4: Effects of message and generation lengths on watermark recovery and text quality using LLaMA3.1-8B-IT on C4. All results are computed over 200 samples.

length increases, the Bit Acc of most methods decreases because the available tokens become insufficient to carry the increasingly dense payload. MuxMark consistently outperforms the other baselines except HeRo. Although HeRo achieves slightly higher Bit Acc, its median PPL remains between 25.4 and 29.8, approximately twice that of MuxMark (13.0-13.3). SegMark remains close to random guessing, while BiMark falls substantially below the 0.5 reference line as the payload increases. These results show that MuxMark provides a favorable recovery–quality trade-off.

Effect of generation length. As shown in Figure 4 (b), increasing the maximum generation length provides more token-level capacity for injecting and aggregating watermark evidence. Consequently, all methods benefit from longer outputs. With a fixed 48-bit message, the Bit Acc of MuxMark increases from 0.787 at $T = 64$ to 0.988 at $T = 512$. Meanwhile, its median PPL decreases from 12.8 to 7.6, indicating that the improved recovery does not require stronger local perturbations as more carrier tokens become available. HeRo achieves the highest Bit Acc (0.997), but its PPL rises from 25.8 to 300.9 as T increases from 64 to 512. Its final accuracy gain over MuxMark is about one percentage point, at nearly $40\times$ the PPL. Thus, the advantage of HeRo is obtained at a substantial cost in text quality, whereas MuxMark approaches near-perfect recovery while preserving low perplexity.

Effect of message length. Figure 4 (c) fixes the $T = 256$ and varies the payload from 16 to 512 bits. Increasing the message length raises the number of bits that must be conveyed by each output token, leading to a consistent decline in recovery accuracy. MuxMark decreases from 0.999 at 16 bits to 0.678 at 512 bits, but remains the strongest method among the quality-preserving baselines. At 512 bits, it outperforms MPAC, StealthInk, XMark, SegMark, and BiMark. HeRo retains a higher Bit Accuracy of 0.753, but its median PPL varies between 39.8 and 89.9 across the evaluated payloads, compared with approximately 8.5-9.6 for MuxMark. The joint Bit Acc and PPL results therefore demonstrate that recovery accuracy alone can obscure substantial differences in generation quality: HeRo favors watermark strength, while MuxMark achieves a more balanced trade-off between payload, recoverability, and text quality.

Watermark injection latency. Figure 5 reports generation time relative to unwatermarked generation. MuxMark incurs a latency ratio of $1.18\times$, corresponding to 18% overhead. This is higher than MPAC, BiMark, XMark, and HeRo (1.02–$1.06\times$), but lower than SegMark ($1.39\times$) and StealthInk ($2.83\times$). These results show that MuxMark improves message recovery and robustness with moderate generation overhead.

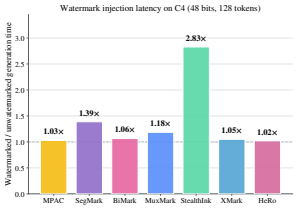


Figure 5: Watermark Latency.

7 Conclusion

By combining binary-channel signals during generation and aggregating weighted votes during recovery, MuxMark allows each token to support multiple message positions. Experiments across model families, payloads, generation budgets, and text perturbations demonstrate improved message recovery while maintaining comparable text quality, with particularly strong gains when recovering long messages from short texts.

AI USE STATEMENT

We used generative AI tools to polish the wording and improve the readability of the manuscript. All AI-assisted revisions were reviewed by the authors to ensure technical accuracy and preserve the intended meaning. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The methodology section describes MuxMark's watermark injection, message recovery, and detection procedures. The experimental setup and appendix document the models, datasets, baseline configurations, and hyperparameters, together with the evaluation protocols and attack settings. The anonymized code repository is available at `https://anonymous.4open.science/r/MuxMark-3EF3`.

REFERENCES

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Improved unbiased watermark for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20587–20601, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1005. URL https://aclanthology.org/2025.acl-long.1005/.

Xuyang Chen, Xiang Li, Yangxinyu Xie, and Qi Long. Selective disclosure watermarking for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Oi1AISOxzt.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Aloni Cohen, Alexander Hoover, and Gabe Schoenbach. Watermarking language models for many adaptive users. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2583–2601. IEEE, 2025.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. Bimark: Unbiased multilayer watermarking for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Zvyb3WAg03.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2023.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Watermarking diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3aBWTYGcaT.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chenxi Gu, Xiaoning Du, and John C. Grundy. SSG: Logit-balanced vocabulary partitioning for LLM watermarking. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36726–36737, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1702. URL https://aclanthology.org/2026.acl-long.1702/.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.226. URL https://aclanthology.org/2024.naacl-long.226/.

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1706–1715, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.98. URL https://aclanthology.org/2024.findings-acl.98/.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45408–45436, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c5b00c5bdcc6fe35907dbcca03d27652-Paper-Conference.pdf.

Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. PMark: Towards robust and distortion-free semantic-level watermarking with channel constraints. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EhDgP69DJG.

Haoyu Jiang, Xuhong Wang, Ping Yi, Shanzhe Lei, and Yilun Lin. Credid: Credible multi-bit watermark for large language models identification, 2025a. URL https://arxiv.org/abs/2412.03107.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27685–27709. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/jiang25j.html.

Ya Jiang, Massieh Kordi Boroujeny, Surender Suresh Kumar, and Kai Zeng. Mirrormark: A distortion-free multi-bit watermark for large language models, 2026. URL https://arxiv.org/abs/2601.22246.

Joeun Kim, HoEun Kim, Dongsup Jin, and Young-Sik Kim. Block-wise codeword embedding for reliable multi-bit text watermarking. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Ym1zKxtv8V.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27469–27500. Curran Associates, Inc., 2023. doi: 10.52202/075280-1195. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/575c450013d0e99e4b0ecf82bd1afaa4-Paper-Conference.pdf.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.268. URL https://aclanthology.org/2024.acl-long.268/.

Hao Li, Yubing Ren, Yanan Cao, Yingjie Li, Fang Fang, Shi Wang, and Li Guo. DualGuard: Dual-stream large language model watermarking defense against paraphrase and spoofing attack. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 23338–23361, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1169. URL https://aclanthology.org/2026.findings-acl.1169/.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6p8lpe4MNf.

Shuliang Liu, Xingyu Li, Hongyi Liu, Dong Fang, Duan Bingchen, Zheng Qi, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for reasoning large language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=T6NVogsXCZ.

Yepeng Liu, Xuandong Zhao, Christopher Kruegel, Dawn Song, and Yuheng Bu. In-context watermarks for large language models. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=fD9YRHazW3.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11724–11735, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.630. URL https://aclanthology.org/2024.acl-long.630/.

Minjia Mao, Dongjun Wei, Zeyu Chen, Xiao Fang, and Michael Chau. Watermarking large language models: An unbiased and low-risk method. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7939–7960, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.391. URL https://aclanthology.org/2025.acl-long.391/.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. WaterMod: Modular token-rank partitioning for probability-balanced LLM watermarking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):32683–32691, March 2026. doi: 10.1609/aaai.v40i39.40546. URL https://ojs.aaai.org/index.php/AAAI/article/view/40546.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for ai-generated text. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025. USENIX Association. ISBN 978-1-939133-52-6.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Yubing Ren, Ping Guo, Yanan Cao, and Wei Ma. Subtle signatures, strong shields: Advancing robust and imperceptible watermarking in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5508–5519, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.327. URL https://aclanthology.org/2024.findings-acl.327/.

Gemma Team. Gemma 3. 2025. URL https://goo.gle/Gemma3Report.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. WaterBench: Towards holistic evaluation of watermarks for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1517–1542, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.83. URL https://aclanthology.org/2024.acl-long.83/.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JYu5Flqm9D.

Yidan Wang, Yubing Ren, Yanan Cao, and Binxing Fang. From trade-off to synergy: A versatile symbiotic watermarking framework for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10306–10322, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.509. URL https://aclanthology.org/2025.acl-long.509/.

Yidan Wang, Yubing Ren, Yanan Cao, and Li Guo. Worldcup sampling for multi-bit llm watermarking, 2026. URL https://arxiv.org/abs/2602.01752.

Jiahao Xu, Rui Hu, Olivera Kotevska, and Zikai Zhang. XMark: Reliable multi-bit watermarking for LLM-generated texts. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14763, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.672. URL https://aclanthology.org/2026.acl-long.672/.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4031–4055, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.224. URL https://aclanthology.org/2024.naacl-long.224/.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

Junlin Zhu, Baizhou Huang, and Xiaojun Wan. QuantileMark: A message-symmetric multi-bit watermark for LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6790–6806, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.308. URL https://aclanthology.org/2026.acl-long.308/.

A BASELINES

For each baseline, we follow the configurations in original papers. The key hyperparameter settings are as follows:

- **MPAC** (Yoo et al., 2024): We adopt the lefthash scheme. The window size is 2, and the proportionγ of green lists is 0.5. A bias $\delta = 2.0$ is added to the logit scores of green tokens. The hash key is 15485863.
- **SegMark** (Qu et al., 2025): We use the RSBH scheme (balanced segment assignment with ECC). The window size is 2, the proportion γ of green lists is 0.5, and a bias $\delta = 2.0$ is added to the logit of green tokens. The salt key is 35317.
- **BiMark** (Feng et al., 2025): The base scaling factor $\tilde{\delta}$ is set to 1.0, and the number of layers d is 10. The proportion γ of green lists is 0.5 and the window size is 2. The values of c_key and bit_idx_key are 530773 and 283519, respectively.
- **StealthInk** (Jiang et al., 2025b): We adopt the probability reweighting scheme with a unit capacity of $m = 1$ bit, corresponding to a red-list proportion of $\gamma = 0.5$. The context window size is $h = 3$. We use SHA-256 as the pseudorandom function and a randomly generated 1024-bit string as the secret key. No additive logit bias is required.
- **XMark** (Xu et al., 2026): We use the LoSo/cTMM configuration with a message block size of $d = 2$ bits, $k = 2$ keyed vocabulary partitions, and a two-token context. Each keyed partition divides the vocabulary into $2^d = 4$ shards and constructs a leave-one-shard-out green list. The intersection of the two green lists therefore has an expected vocabulary proportion of $(3/4)^2 = 9/16 = 0.5625$. A logit bias of $\delta = 2.0$ is applied to tokens in this evergreen intersection.
- **HeRo** (Chen et al., 2026): We use hash-seeded hierarchical vocabulary routing with a context width of $h = 4$ and two routing levels, each carrying 2 bits, i.e., `bits_per_layer` $= [2, 2]$. Thus, each message segment contains 4 bits. The first level partitions the candidate vocabulary into 20 contiguous chunks (`split_schedule` $= [20]$), selects a chunk by keyed Gumbel-Max sampling, and the second level selects a token within that chunk using a separate key associated with the remaining 2-bit symbol.

B DATASETS

We follow previous work (Tu et al., 2024) to evaluate our multi-bit watermark method on the following datasets:

- **C4** (Raffel et al., 2020) dataset is a large-scale, high-quality English pretraining corpus constructed by Google from Common Crawl. After extensive cleaning and filtering to remove non-linguistic, low-quality, and duplicate content, it yields roughly 750 GB of clean English text. We use the processed version in `https://huggingface.co/datasets/allenai/c4`.
- **CNN_DailyMail** (Hermann et al., 2015) dataset is a large-scale English news corpus containing over 300,000 unique articles written by journalists from CNN and the Daily Mail. The current release supports both extractive and abstractive summarization. We collect 200 samples and prompt the model to produce a one-sentence summary, with max_new_tokens = 64 and min_new_tokens = 32 (**long input, short output**), and embed a **16-bit** message into each generated summary. Details of the dataset can be found at: `https://huggingface.co/datasets/abisee/cnn_dailymail`.
- **ELI5** (Fan et al., 2019) dataset is a long-form QA dataset sourced from the Reddit community "Explain Like I'm Five." It contains 270k diverse questions that require multi-sentence, explanatory answers supported by web evidence. We get 200 data points and set max_new_tokens = 256 and min_new_tokens = 64 (**short input, long output**), and embed a **32-bit** message into each generated answer. We use the processed subset available at `https://github.com/THU-KEG/WaterBench/blob/main/data/WaterBench/2-1_longform_qa.jsonl`.

C METRICS

All evaluation metrics used in our experiments are described in detail below:

- **Bit Accuracy** measures the proportion of correctly extracted bits across all samples. Let $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_L^{(j)})$ denote the embedded message of length L for the j-th sample, and $\hat{\mathbf{m}}^{(j)} = (\hat{m}_1^{(j)}, \ldots, \hat{m}_L^{(j)})$ be the corresponding extracted message. The Bit Accuracy over n samples is defined as

$$\text{Bit Acc.} = \frac{1}{nL} \sum_{j=1}^{n} \sum_{i=1}^{L} \mathbb{I}\big[\hat{m}_i^{(j)} = m_i^{(j)}\big] \tag{12}$$

  where $\mathbb{I}[\cdot]$ denotes the indicator function. It reflects fine-grained bit-level decoding performance, as shown in Table 1.

- **Exact-match Accuracy** is the proportion of samples for which the entire embedded message is recovered correctly. Using the notation above, it is defined as

$$\text{Exact Acc.} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\big[\hat{\mathbf{m}}^{(j)} = \mathbf{m}^{(j)}\big]. \tag{13}$$

  A sample is counted as correct only if all L bits are recovered without error. This metric complements Bit Accuracy by evaluating message-level recovery.

- **Perplexity (PPL)** measures how well a language model predicts a given text. For a sequence $x_{1:T}$:

$$\text{PPL} = \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \log p(x_t \mid x_{<t})\right) \tag{14}$$

  Lower perplexity implies more confident and accurate language modeling. We report median PPL rather than mean PPL, as it provides more stable estimates and is less sensitive to extreme values (Jiang et al., 2025b).

- **ROUGE Score** (Lin, 2004) measures overlap between generated and reference text. Among its variants, we use ROUGE-L, which computes the longest common subsequence (LCS) between the candidate and reference, capturing sentence-level structural similarity beyond contiguous n-gram overlap.

- **ΔGPT5 Score** leverages gpt-5.6-sol's xhigh mode to do *Paired LLM-based QA evaluation.* To reduce calibration drift in independently conducted LLM evaluations, we adopted a sample-level, anonymized paired evaluation protocol. For each watermarking method and target language model, we sampled 200 QA instances. For every instance, the watermarked response and its corresponding unwatermarked response were presented to `gpt-5.6-sol` with `xhigh` reasoning effort as two anonymous candidates, denoted as Candidate A and Candidate B. Both candidates were evaluated within the same record using the same question and reference answer. The evaluator was not provided with the watermarking method, response provenance, or which candidate contained a watermark. To control for position bias, the watermarked response was assigned to Candidate A for exactly 100 instances and to Candidate B for the remaining 100 instances, with the assignment randomly shuffled. Each candidate was evaluated according to five criteria: (1) fluency and naturalness, (2) adequacy and factual agreement with the reference answer, (3) coherence, (4) relevance to the question and key reference information, and (5) stylistic consistency with the reference answer. The evaluator assigned an integer score from 1 to 10 for each criterion, and the arithmetic mean of the five criterion scores was used as the final QA score. Consequently, the final score ranged from 1 to 10 in increments of 0.2. The evaluator scored the two candidates independently and was explicitly instructed neither to force a winner nor to force a tie. For each model–method combination, we report the mean scores of the watermarked and paired unwatermarked responses. We define the paired quality difference as

$$\Delta = \frac{1}{N} \sum_{i=1}^{N} \left(s_i^{\text{wm}} - s_i^{\text{no-wm}}\right), \qquad N = 200,$$

where s_i^{wm} and $s_i^{\text{no-wm}}$ denote the scores of the watermarked and unwatermarked responses for the same QA instance, respectively. We computed pointwise 95% confidence intervals for Δ using paired bootstrap resampling over QA instances with 10,000 bootstrap replicates.

D Attack Settings

To evaluate the robustness of the watermark, we design the following various attack scenarios:

- **Word-D (ratio = ρ)**: Randomly deletes a proportion ρ of words from the watermarked text.
- **Word-S-Dict (ratio = ρ)**: Randomly replaces a proportion ρ of words with their synonyms based on the WordNet (Miller, 1995) lexical dictionary.
- **Word-S-BERT (ratio = ρ)**: Randomly substitutes a proportion ρ of words with context-aware synonyms generated by a BERT-based (Devlin et al., 2019) model.
- **Copy-Paste ($n - \rho$)**: Randomly splits the watermarked text into n segments and inserts them into non-watermarked text, such that the inserted non-watermarked content accounts for a total proportion ρ.
- **Translation (en–zh)**: Translates the watermarked text from English to Chinese and then back to English using a fine-tuned T5 translation model: `https://huggingface.co/utrobinmv/t5_translate_en_ru_zh_small_1024`.
- **Rephrase (DIPPER-1)** (Krishna et al., 2023): Rephrases the text using the DIPPER model with controlled lexical diversity and word order diversity (lex_diversity=ρ_1 order_diversity=ρ_2, max_new_tokens=256, sent_interval=1, top_p=0.75).
- **Rephrase (DIPPER-2)** (Krishna et al., 2023): Rephrases the text using the DIPPER model under alternative settings of lexical and order diversity (lex_diversity=ρ_1' order_diversity=ρ_2', max_new_tokens=256, sent_interval=1, top_p=0.75).

E Hyper-parameters Analysis

Effect of k_{max} and δ. We analyze the effect of the maximum number of encoded bits per token, $k_{\max}$, in Figure 6. When $k_{\max}$ increases from 2 to 8, Bit Acc improves from about 0.899 to 0.913, showing that allowing each token to carry more bit positions can improve message recovery. However, further increasing $k_{\max}$ to 16 slightly reduces Bit Acc, suggesting that overly dense bit allocation may introduce less reliable votes. We therefore use $k_{\max} = 6$ as the default setting, which achieves the best recovery accuracy while keeping PPL stable. Experiments on δ are listed in Figure 3 and Figure 6. As δ increases, MuxMark achieves higher Bit Acc at the cost of higher PPL, showing a clear trade-off between watermark strength and text quality. We choose $\delta = 1.9$ as the default setting, which provides a strong recovery performance while keeping PPL within a moderate range.

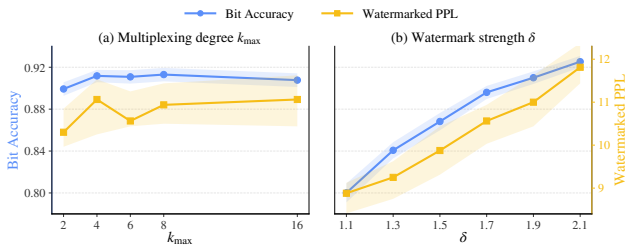


Figure 6: Effect of $k_{\max}$ and δ.

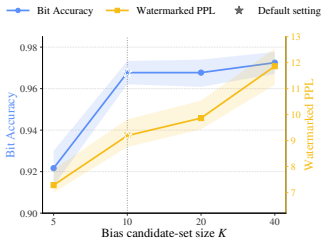


Figure 7: Effect of Top-K.

Effect of the bias candidate-set size top-K. During generation, MuxMark restricts watermark biasing to the top-K tokens under the original next-token distribution. A small K preserves text quality by preventing low-probability tokens from receiving watermark bias, but it also limits the number of candidate tokens available for carrying the multiplexed signal. We evaluate $K \in \{5, 10, 20, 40\}$ on C4 using LLaMA3.1-8B-IT, with a 48-bit message, a maximum generation length

of 256 tokens, and 100 samples per setting. As shown in Figure 7, increasing K from 5 to 10 substantially improves bit accuracy from 0.9217 to 0.9677, while the watermarked PPL increases from 7.2800 to 9.1875. Further enlarging the candidate set provides only marginal recovery gains: $K = 20$ achieves the same bit accuracy of 0.9677 with a higher PPL of 9.863, whereas $K = 40$ improves bit accuracy by only 0.0048 over $K = 10$ but raises PPL to 11.8631. All evaluated settings achieve an AUROC of 1.0, indicating that detection remains stable across the tested range. These results identify $K = 10$ as the knee point of the recovery–quality trade-off, providing strong message recovery without unnecessarily extending the watermark bias to lower-probability candidates. We therefore adopt $K = 10$ in the main experiments. Shaded regions in the figure denote 95% bootstrap confidence intervals.

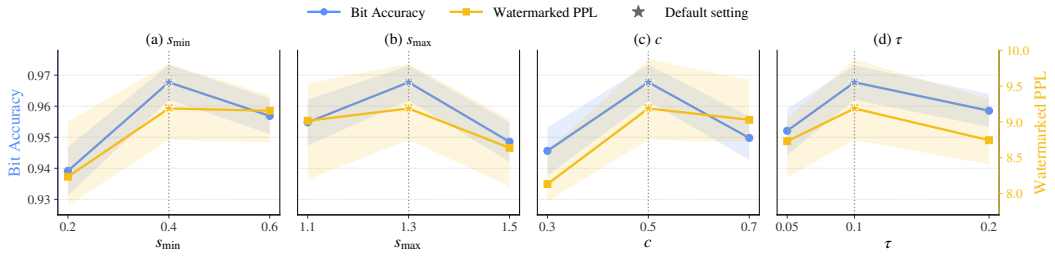


Figure 8: Sensitivity to entropy-scaling hyperparameters.

Sensitivity to entropy-scaling hyperparameters. We analyze the four hyperparameters of the entropy-dependent scaling function

$$s(H) = s_{\min} + (s_{\max} - s_{\min})\sigma\left(\frac{\bar{H} - c}{\tau}\right), \tag{15}$$

where $s_{\min}$ and $s_{\max}$ control the minimum and maximum watermark strengths, c determines the transition center, and τ controls the smoothness of the transition. We vary one parameter at a time while fixing the others to the default configuration $(s_{\min}, s_{\max}, c, \tau) = (0.4, 1.3, 0.5, 0.1)$. Experiments are conducted on C4 using LLaMA3.1-8B-IT with a 48-bit message, a maximum generation length of 256 tokens, and 100 samples per configuration. As shown in Figure 8, all configurations achieve an AUROC of 1.0, indicating that watermark detection remains stable throughout the examined range. Nevertheless, the entropy parameters affect the trade-off between message recovery and perplexity. The default configuration achieves the highest bit accuracy of 0.9677 with a watermarked PPL of 9.1875. Reducing $s_{\min}$ to 0.2 lowers the PPL to 8.2337 but decreases bit accuracy to 0.9392, whereas increasing it to 0.6 yields a bit accuracy of 0.9569. Similarly, moving $s_{\max}$ away from 1.3, shifting the transition center from $c = 0.5$, or using a sharper or smoother transition than $\tau = 0.1$ consistently reduces message recovery, although some settings obtain lower perplexity. Overall, MuxMark remains effective across the tested parameter ranges, while the selected configuration provides the strongest message recovery with only a modest perplexity cost, supporting its use in the main experiments.

F AVERAGE OBSERVATIONS PER BIT.

Observation efficiency. Table 5 compares the observation efficiency of MuxMark with the baselines using their actual outputs from the main experiment. Because conventional methods produce approximately one token-level decoding observation per scorable output token, they obtain only 5.25–5.29 observations per payload bit under a 256-token budget and a 48-bit payload. In contrast, MuxMark allows each token to provide evidence for multiple message positions. With $k_{\max} = 6$, the realized average multiplexing degree is 3.456, resulting in 18.28 observations per bit (95% CI: $[18.195, 18.365]$). This corresponds to a $3.46\times$–$3.48\times$ increase in observation density over the baselines under the same token budget. The increased density supplies repeated evidence for each message position and therefore improves the utilization of short outputs, especially when the payload is long. For the block-based baselines, a token may jointly encode several bits, but this constitutes a single multi-class observation shared by the corresponding block rather than multiple separately

Table 5: Average number of token-level decoding observations per bit. Results are measured on C4 using LLaMA3.1-8B-IT with a 48-bit message, a maximum generation length of 256 tokens, and 200 texts. Confidence intervals are obtained by bootstrapping over texts.

Method	**Avg. observations/bit**	**95% CI**
MPAC	5.289	$[5.287, 5.290]$
SegMark†	5.254	$[5.222, 5.271]$
BiMark	5.290	$[5.289, 5.291]$
StealthInk	5.270	$[5.269, 5.271]$
XMark†	5.290	$[5.289, 5.291]$
HeRo†	5.250	$[5.248, 5.252]$
MuxMark	**18.280**	$[\mathbf{18.195}, \mathbf{18.365}]$

The statistic counts separately reconstructed token-level evidence items and is computed as $\overline{O} = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_t q_{i,t}}{b}$. For conventional methods, $q_{i,t} = 1$ for each scorable token, whereas MuxMark uses $q_{i,t} = k_t$ because a token produces separately recoverable binary evidence for k_t message positions. Internal hash keys or partition projections targeting the same payload unit are not counted repeatedly. †SegMark, XMark, and HeRo produce joint observations for multi-bit symbols or blocks; each such observation is counted once rather than being treated as multiple independent binary observations.

reconstructed binary observations. MuxMark instead increases the number of explicit binary observations while retaining binary vocabulary partitions, avoiding the rapidly growing decision space associated with large multi-bit symbols.

G WATERMARKING EXAMPLES

We present four qualitative examples of MuxMark on C4. All examples use LLaMA3.1-8B-IT, a maximum generation length of 256 tokens, and no attack. Examples 1–2 show exact message recovery, whereas Examples 3–4 show partial recovery. The watermarked and unwatermarked texts are continuations of the same prompt. For readability, the displayed continuations are truncated at the last complete sentence shown, evaluation was performed on the original full outputs. The message and decoded message are copied from the evaluated output files.

Table 6: [Example 1] Exact recovery of a 24-bit message (index: 6520).

Prompt: Interviews are on in Dublin at the end of March. ONE OF CANADA'S hair salons is recruiting and they'

Unwatermarked text

re offering free interviews at the end on March in Dublin. Davenport Hair Salon, a well-known destination for beauty enthusiasts in Toronto, is currently seeking talented hair stylists and colourists to join their team. They will be holding open interviews for the positions from March 28th to 30th at their salon on Dundee Street. Davenport Hair Salon has been a hair salon of choice for people in the Greater Toronto area since its inception in 1976. They're known for their expert team and top quality products that will make you look and feel like the best version of yourself. Their team is dedicated to providing personalized service, creating unique and bespoke looks and providing exceptional customer care. They believe in investing in their team members, providing ongoing education and support to ensure their stylists and colorists are on the cutting edge of the latest trends and techniques. If you're a passionate hairstylist or colourist with a knack for creating beautiful, on-trend looks, apply now!

Watermarked text

re looking for stylists to join their team. Kozmo, a well-established company with locations on Dublin's Dame Street, Stoneybatter and Ranelagh, is currently on the hunt for passionate and skilled stylists to add to their talented team. As well as Dublin, Kozmo also has locations in Dundalk and Laois, and has been growing since it was founded back in 2003. If you're looking for a challenging and rewarding role as a hairstylist, with a fun and dynamic workplace to boot, you can submit an application now to be considered. A passion for delivering exceptional customer service and creating unique looks to make their clients look and feel amazing. Excellent communication skills to understand each client's needs and preferences.

Message = decoded: `011001100010100110110101` **Bit Acc:** 1.0000 (24/24)

Table 7: [Example 2] Exact recovery of a 48-bit message (index: 2513).

Prompt: The morning before I attended the screening of the new X-Men movie, I was cleaning out an old box of mementos and came across

Unwatermarked text

an old photo that I'd taken at a comic book store. In it, I'm standing in front of a shelf lined with X-Men comics, wearing a homemade Wolverine mask. It was taken around 1983. I was 12 years old. I have no idea who the other kids in the picture are. I don't even remember how I got the mask or who else was with me that day, and we clearly weren't in a group. It's just a faded Polaroid of a pre-teen me proudly striking a pose as a young mutant, grinning from ear to ear, with my arms crossed and my claws at the ready. Who knew 36 years later, I'd be writing about a movie starring a character I'd once emulated? And I never, ever thought I'd actually become a published author. And here I am, reflecting on the passing of time, nostalgia and how the things that inspire us as kids can often shape the direction of our lives in ways we can hardly imagine.

Watermarked text

my ancient, faded X-Men action figure collection. I dusted them all off, placed them out for display, and I spent a few minutes reminiscing about a simpler era. These little figures represent a moment in time, a nostalgia for something carefree, exciting and adventurous. They remind me of the countless Saturday afternoons spent playing X-treme X-Men and X-Wars, creating imaginative scenarios where I and my friends took on the roles of our favorite heroes and villains. Those hours of entertainment brought so much joy to my life during my youth. Fast-forward to now and I find myself reminiscing about those bygone days. These toys that were once a source of endless entertainment now seemed a reminder that times are different today. Gone are the lazy afternoon playtime days where all our care and worries seemed a world away. As an adult now I face the weight of responsibilities, bills, stress, and uncertainty. These toys seem out of place. They remind us all of when our days consisted of endless adventures rather than constant adulting.

Message = decoded: 000111010000001110000110101011001000001010110000
Bit Acc: 1.0000 (48/48)

Table 8: [Example 3] 24-bit recovery with one bit error (index: 424).

Prompt: Home Procurement News Why is lean accounting necessary in the supply chain? Why is lean accounting necessary in the supply chain? Knowing how

Unwatermarked text

to apply value stream mapping to drive lean accounting How to conduct a value stream mapping workshop Lean Accounting in the Supply Chain: Part 1, Value stream mapping, a Lean methodology used to map out the flow of value-creating processes, is a crucial element in supply chain management. It provides a clear and concise representation of the current process flow and future improvements. This article will explore why lean accounting in the supply lead time management is necessary to improve supply chain efficiency.

Watermarked text

much to inventory, when to inventory, where and when deliveries take place and how often they will take place and where they go, is a must have data in the logistics of running a smooth and effective supply chain management system. The goal is to minimize stock holding in the warehouse and to ensure that stock levels in the warehouse should match with what the company needs at the moment. The need of having to keep track of inventory also means to keep track where your product has been placed, when it is received, when was it sent to and how long has it been in the warehouse or with the shipping company. All the information should be recorded accurately.

Message: 011100000010000111011001
Decoded message: 011100000010000111011101
Bit Acc: 0.9583 (23/24) **Mismatched bit position:** 22

Table 9: [Example 4] 48-bit recovery with three bit errors (index: 2353).

Prompt: Giving gifts just ain't my bag. Christmas music makes me gag. The critter exaggerateth not: Though the season is supposed

Unwatermarked text

to be jolly, some people just can't help but think the holidays are actually one big, long, commercialized nightmare. If that's the case, I'm with them. I mean, who needs Christmas songs that involve "reindeer," "frosty," or "snow?" I much prefer the ones that celebrate peace, joy, and giving - not just a day to buy stuff and consume. So, my musical holiday picks for you are some of those types of tunes. Think of these songs as a respite from all the yuletide yack. Bing Crosby and David Bowie's "Peace on Earth/Little Drummer Boy," for example, is a beautiful duet that focuses more on the spirit of the season than on the saccharine aspects of it. Written by Crosby and Ian Tyson, it's a simple, heartfelt ballad that highlights the importance of kindness and compassion.

Watermarked text

to be a time of joy, some people, especially those with mental health issues, can find it a really difficult time of year. For me, the whole business about "getting into the spirit" and all the pressure to make everything "festive" is overwhelming. It's a time when the expectations are sky high and feelings of sadness are low. For those of us on this wavelength, the music that used to be a source of pleasure has instead become an ear-sore. The commercialization and materialism seem to be the greatest turnoff for a lot of people during this time of year, including those with depression and anxiety. There's an expectation that we should all be merry and bright, when for many of us, it is anything but. It's an expectation placed upon us that can lead to feelings of isolation and guilt. As we go about planning the season, we need to think carefully. What really makes us happy and what we can enjoy? What will allow others to enjoy the season without feeling burdened with expectations? For some people, the music might not be the answer; the decorations, the parties or the pressure to spend. Maybe for you, there are other things like baking, cooking or spending quality time with family and friends.

Message: 010111100100011110100010011011010010110111101110
Decoded message: 010011100100011110100010011010010000110111101110
Bit Acc: 0.9375 (45/48) **Mismatched bit positions:** 4, 30, 35